Feature-Guided Upsampling for Semantic Segmentation

Raymond Weiming Chan

Abstract—Encoder-decoder architectures, particularly the U-Net and its variants, are foundational for semantic segmentation tasks. The encoder strives to extract rich semantic features, and the decoder attempts to recover spatial resolution. Traditional upsampling methods tend to be either semantically agnostic, computationally expensive, or result in blurred object boundaries or lost details. We propose Feature-Guided Upsampling (FGU), a novel upsampling mechanism designed for the decoder portion of U-Net architectures. FGU improves upon regular skip connections by incorporating feature projection and attention. This mechanism leverages high-resolution spatial feature maps from the encoder to dynamically guide the upsampling process, with the goal of filtering background noise while enhancing structural fidelity. We integrate FGU into a standard U-Net backbone and evaluate its performance on the CamVid dataset. Our experiments demonstrate that FGU outperforms baseline, achieving a 2.4% increase in Dice score under the same training settings. Despite the additional model complexity over baseline, no extra hyperparameter adjustments were needed to achieve these results. Qualitative results further indicate that our method outperforms baseline in boundary adherence, particularly in areas involving thin objects and shadows.

1 Introduction

Semantic segmentation is the task of assigning a classification label to every pixel in an image. It sees use in a wide array of applications such as autonomous driving, robotic navigation, medical image analysis, and satellite imagery interpretation [1], [2], [3], [4]. Many architectures have been developed, using U-Net as a foundation, to meet the demand for better pixel-level accuracy, particularly for tasks requiring precise boundaries.

The dominant paradigm for this task is the encoder-decoder architecture, exemplified by the U-Net [5] and Fully Convolutional Networks (FCNs) [6]. In these architectures, the encoder path progressively reduces spatial resolution to extract high-level semantic features, while the decoder path attempts to reconstruct the spatial resolution to generate the final dense prediction. While encoders have significantly benefited from advances in backbone architectures like ResNet [7], the decoder's upsampling process remains a critical bottleneck. The challenge lies in effectively recovering fine-grained spatial information that was lost during the downsampling operations of the encoder.

Upsampling approaches are generally categorized to be fully deterministic or learnable. Fully deterministic methods like bilinear interpolation or nearest-neighbor interpolation are computationally efficient, but often fail to preserve fine details, resulting in overly smoothed predictions. Learnable methods typically involve using neural networks. For example, transpose convolution was used in the original U-Net, but it suffers from the generation of checkerboard artifacts [5]. It also requires additional search for optimal hyperparameters since the model performance is sensitive to learning rate. To mitigate the loss of spatial detail, the standard U-Net uses skip connections, but these skip connections are simple concatenation. This approach is naive, and assumes that all spatial information from the encoder is relevant for reconstruction. In practice, encoder features often contain background noise or irrelevant textures that can "confuse" the decoder. The semantic gap between the shallow, low-level features of the encoder and the deep, high-level features of the decoder further exacerbates this issue.

We address these limitations by proposing Feature-Guided Upsampling (FGU), a module designed to replace the upsampling operation in the decoder of a U-Net-style network. Unlike standard skip connections that passively forward information, FGU actively filters and selects spatial features.

We hypothesize that the recovery of spatial resolution should be guided by the correlation between the encoder's detailed structure and the decoder's semantic context. To achieve this, we apply spatial attention and channel attention [8] after the feature fusion step. Before feature fusion, we also perform feature projection by convolution to address the semantic gap. In a model with transpose convolution, the transpose convolution step attempts to increase resolution and interpret features at the same time. Our model decouples the problem by letting bilinear upsampling handle geometry, and convolution handle semantics. This leads us to hypothesize that our model will be much more robust to learning rate and other hyperparameters. For example, for transpose convolution, if the learning rate is too high, the upsampling introduces jaggedness in the result.

2 Related Work

Major improvements were made in semantic segmentation with the introduction of FCNs [6], which replaced fully connected layers with convolutional layers. FCNs suffered from coarse outputs due to excessive downsampling. The U-Net architecture [5] addressed this by introducing an encoder-decoder structure connected by skip connections, using high-resolution features from the encoder to assist in recovering spatial details in the decoder stages. The U-Net has become the gold standard for medical imaging, but it

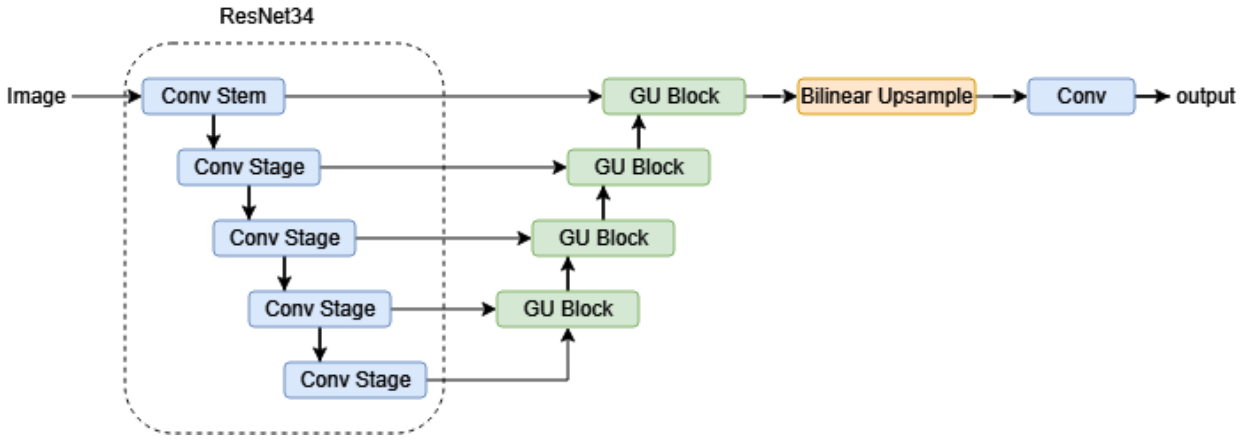

Fig. 1. High-level architectural overview

still suffers from the ”semantic gap” [9]. The semantic gap is the difference in context of the encoder and the decoder. Features from the encoder are low-level, representing edges and textures, while features from the decoder are high-level that represent object-level information. Naively concatenating these features, as in the original U-Net, leads to feature misalignment and noise propagation, motivating the need for a smarter fusion of the encoder features and the decoder features.

One avenue in combatting this issue is through attention mechanisms. Attention mechanisms allow networks to focus on relevant features while suppressing irrelevant ones. Squeeze-and-Excitation (SE) networks [10] introduced channel attention, which uses global average pooling to ”squeeze” spatial information into a channel descriptor and then applying a small gating MLP to ”excite” each channel with learned importance weights. Building on this, the Convolutional Block Attention Module (CBAM) [8] combined Channel Attention with Spatial Attention sequentially to refine features in both dimensions. CBAM was originally proposed for feature extraction in encoders, our work explores its utility as a post-fusion refinement tool in the decoder.

There are previous works that integrated attention into U-Net architectures to improve segmentation, such as Attention U-Net [11], which introduced additive attention gates on the skip connections. Their method filters the skip connection before concatenation to suppress background regions. Other networks like FGA-Net [12] employ a top-down guidance approach. Attention is applied before and during fusion, and the encoder is customized. Our work explores the effectiveness of post-fusion attention (on a pre-trained encoder), which theoretically filters noise from both encoder and decoder simultaneously. Our approach is ”lateral” rather than top-down, as we use encoder features to guide the upsampling of the decoder features.

On effectively fusing encoder and decoder features, methods like RefineNet [13] demonstrated that multipath refinement blocks could effectively combine high and low-level features using residual connections. Guided Upsampling Networks (GUNs) [14] puts emphasis on inference speed and uses a single convolution to guide the upsampling, but it has a similar pitfall to that of transpose convolution - noise handling and upsampling are handled together by single neural layers.

3 Proposed Method

Our architecture adopts a U-Net like encoder-decoder structure. Our FGU block is designed to bridge the semantic gap between high-level feature extraction and high-resolution mask reconstruction.

Using a pretrained encoder ”backbone” is common practice in the context of practically building U-Net style networks [15]. We choose ResNet34 as the encoder backbone to achieve an optimal balance between feature extraction capability and computational efficiency. Unlike deeper variants such as ResNet50 or ResNet101, which are prone to overfitting on limited datasets, ResNet34 provides sufficient depth to capture complex semantic features while maintaining a parameter count that facilitates robust generalization.

Each of the four FGU blocks receive two features each, the low-resolution semantic features F_{low}, and the high-resolution guidance features F_{high}. Direct concatenation of F_{low} and F_{high} is suboptimal due to the discrepancy in their features. F_{high} contains low-level edge information, while F_{low} contains high-level semantic context. To resolve this, we first align the spatial dimensions via bilinear upsampling on F_{low}. Simultaneously, we pass the guidance features F_{high} through a non-lineaer projection head consisting of two stacked 3×3 convolutions followed by batch normalization and ReLU. F_{low} is also passed through one convolution followed by batch normalization and ReLU. The idea behind this feature projection step is that it is a learnable adaptation layer. The pre-fusion convolutions may enable better contextual analysis of high-frequency details, e.g. is this sharp pixel a part of a continuous line or a speck of noise. ReLU is also used to introduce non-linearity, allowing the guidance to learn complex relationships even before fusion, e.g. only pass this edge if it is vertical and surrounded by a specific texture.

Following the projection, the features are concatenated and fused via a 3×3 convolution to reduce channel dimensionality. This fused feature map $F_{fused} \in \mathbb{R}^{C\times H\times W}$ contains structural details and residual background noise

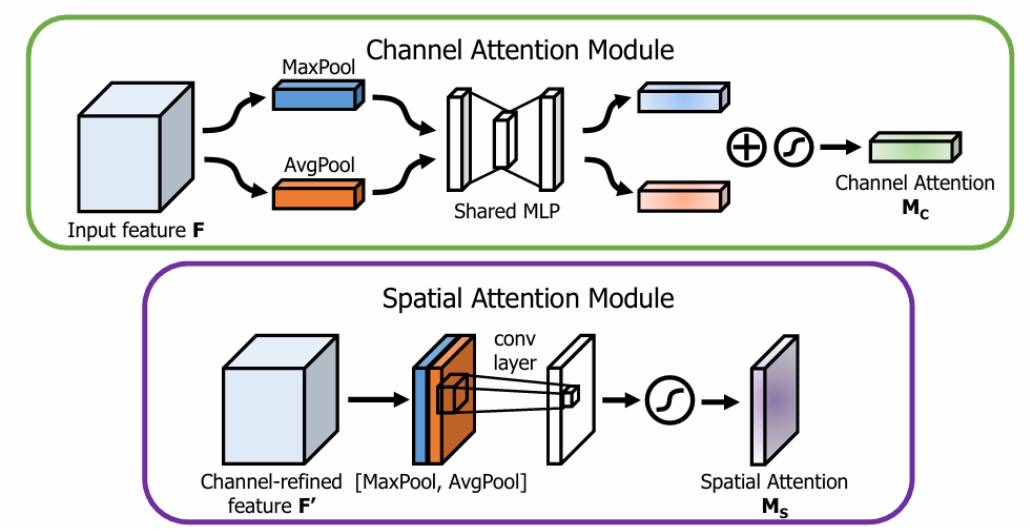

Fig. 2. Channel and Spatial Attention Modules from CBAM [8]

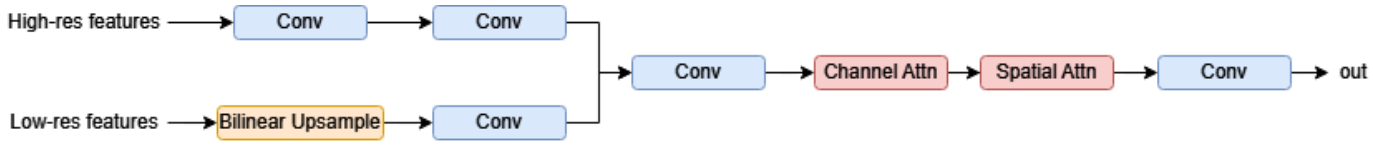

Fig. 3. Our Guided Upsampling block

from the encoder. To filter this noise, we integrate CBAM [8] as a post-fusion gating mechanism. Unlike standard Attention U-Nets which gate inputs before fusion, our approach applies attention after fusion to refine the combined representation. While pre-fusion methods filter skip connections based on partial information, they risk discarding high-frequency details that may not appear to be important until they are combined with the decoder's deep semantic context. By applying attention after concatenation, the relationship between the encoder features and the decoder features are better utilized.

CBAM involves first applying a channel attention map $M_c \in \mathbb{R}^{C\times1\times1}$ and then a spatial attention map $M_s \in R^{1\times H\times W}$.

$$F'_{fused} = M_c(F_{fused}) \otimes F_{fused}$$

$$F''_{fused} = M_s(F'_{fused}) \otimes F'_{fused}$$

In a deep network, each channel represents a specific feature detector [16], and channel attention focuses on "what" is meaningful given an input image. The spatial dimension of the input feature map is squeezed to compute channel attention efficiently, average-pooling is used to aggregate spatial information. Also, max-pooling is used to gather clues about distinctive object features. The original CBAM work has empirically shown that using both features together improves representation power. Both descriptors are forwarded to a shared MLP of one hidden layer and sigmoid activation.

The spatial attention module comes into play to determine which pixels in the image are important. After channel attention, we know which features are important, but those features are still spread across the entire spatial grid. Spatial attention is computed by first applying average-pooling and max-pooling operations along the channel axis and concatenating them to generate a feature descriptor. Applying pooling operations along the channel axis is shown to be good for identifying useful areas [17]. The descriptors are forwarded to a convolutional layer and then sigmoid activation. The original CBAM work also notes that applying channel attention first is slightly more beneficial than parallel or spatial attention first arrangements.

The network is implemented in PyTorch. The segmentation_models_pytorch library is used for the encoder backbone. The decoder blocks are constructed dynamically to match the channel dimensions of the ResNet34 stages (512, 256, 128, 64). The final segmentation mask is generated via a 1×1 convolution followed by a bilinear interpolation step to restore the output to the original input resolution.

4 Experimental Results

To validate the efficacy of the proposed U-Net with feature-guided upsampling and attention, we compared its performance against a standard baseline U-Net with bilinear upsampling. The baseline model employs the identical ResNet34 encoder but uses standard decoder blocks consisting of simple bilinear upsampling followed by concatenation and convolution, lacking the proposed guidance processing and post-fusion CBAM attention mechanisms. Experiments were run on the CamVid dataset [18]. The poles and complex shading conditions in this dataset for autonomous driving are of particular interest. Data augmentation was applied: horizontal flip, resized crop, and color jitter.

To isolate performance gains attributed strictly to the architectural innovations, both the proposed model and the baseline were trained under identical experimental conditions. The encoder parameters had a learning rate of 1e-5. The low learning rate prevented the backbone from deviating too much from the pretrained weights while still adapting to the new data. The decoder parameters had a learning rate of 1e-3. AdamW optimizer with weight decay of 5e-2 was used. A ReduceLROnPlateau scheduler was also used. Both models were trained for 100 epochs using loss objectives Dice loss and cross entropy loss.

Table 1 summarizes the quantitative performance on the test set. Despite the identical training configuration, the proposed method achieved a consistent improvement across all segmentation metrics. The quantitative performance gap highlights that the standard bilinear upsampling approach struggles to effectively fuse the semantic content of the decoder with the spatial details of the encoder. By introducing our guided upsampling block, our model successfully resolved these ambiguities without requiring any hyperparameter tuning. More work needs to be done to investigate if major quantitative improvements can be achieved through hyperparameter tuning.

A critical challenge in outdoor scene segmentation is distinguishing between physical object boundaries and photometric edges caused by shadows. As illustrated in Figure 4, the baseline model exhibits sensitivity to strong cast shadows. In this specific instance, the baseline model misinterprets the shadow boundary across the traffic island as a physical separation, resulting in the misabelling. In contrast, our proposed model successfully segments the traffic island as a single object unaffected by the shadow. This demonstrates the efficacy of the feature-guided upsampling strategy. We hypothesize that the guidance convolutional layers learn to extract texture-invariant structural features, effectively filtering out luminance variations before fusion. Furthermore, the post-fusion attention mechanism allows the deep, high-level context (which recognizes the island's geometry) to override the misleading low-level intensity gradients caused by the shadow. This result confirms that our architecture is not merely matching pixel intensities but is learning robust, illumination-invariant structural representations.

An advantage that our model provides is the correct labelling of thin objects such as poles. As illustrated in Figure 6, the baseline model often fails to detect these objects. The weak signal of the thin object is often overwhelmed by the dominant background features during simple concatenation. By the time the signal hits the bottleneck, the pole pixels has likely been averaged out into the background pixels. On the other hand, our feature-guided attention module successfully reconstructs the pole, likely thanks to the spatial attention mechanism. By directly using the high-resolution guidance signal, the spatial attention module produces a confidence mask that highlights important edges. A similar situation can be seen in Figure 5, where an archway in the distance (small in the image) is lost by simple bilinear upsampling, but is recovered by our model.

A significant finding of this study is the optimization stability of the proposed upsampling block. Typically, introducing additional architectural complexity – in this case attention mechanisms, projection heads, and deep fusion layers – leads to need of extensive hyperparameter tuning to achieve convergence. Our proposed model achieved superior performance without any modification to the hyperparameters used to train baseline. Despite the increased parameter count and depth in the decoder, the model did not suffer from gradient instability or convergence difficulties. The fact that our model outperforms the baseline, which uses non-learnable upscaling, shows that our technique is robust. This is further evident when looking at the similarity of the loss curves (Figure 4 and Figure 5).

Limitations: While the proposed Feature-Guided Attention architecture demonstrates clear advantages in boundary precision and training stability, it is important to acknowledge the trade-offs inherent in its design. The most immediate limitation is the increase in computational cost. Unlike standard U-Net architectures where skip connections are simple memory-copy operations, our approach requires

TABLE 1
Comparison of results

	mIoU	Pixel Accuracy	Dice Score
Baseline	0.4489	0.9096	0.5485
Ours	0.4578	0.9134	0.5646

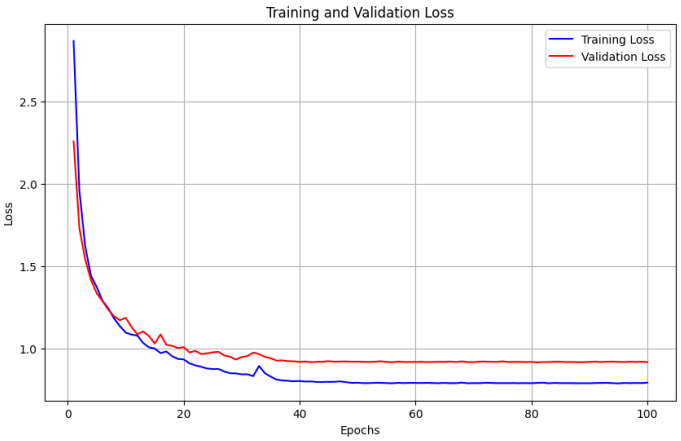

Fig. 4. Loss curve of baseline model

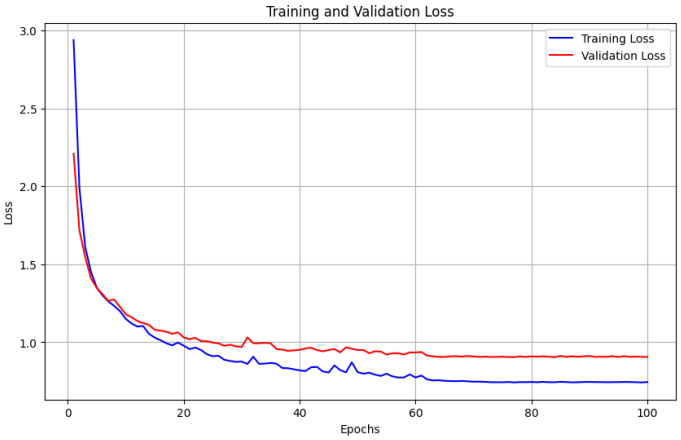

Fig. 5. Loss curve of our model

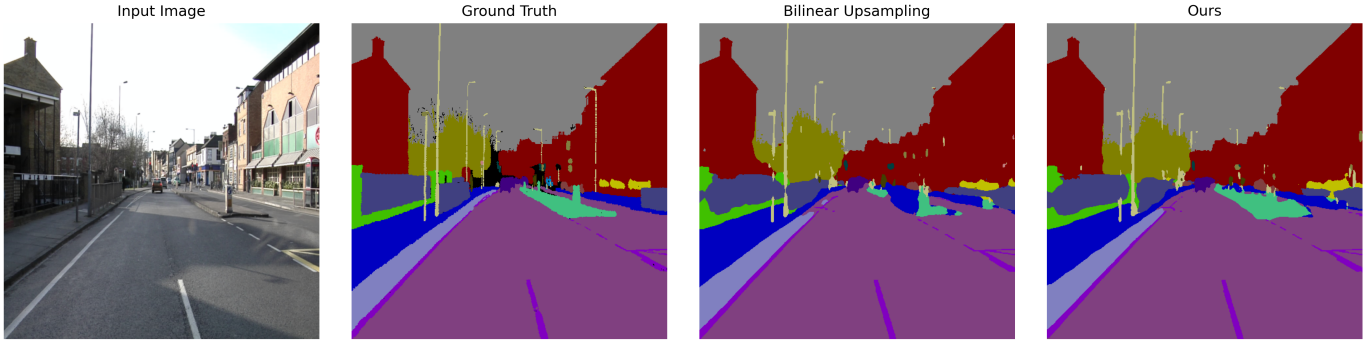

Fig. 6. Ours achieves better results on shaded objects

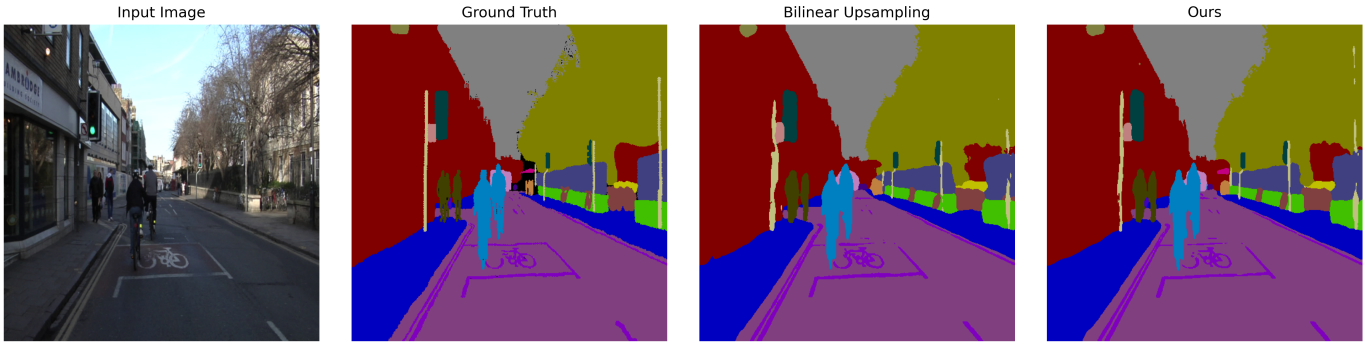

Fig. 7. Ours achieves better results on far objects

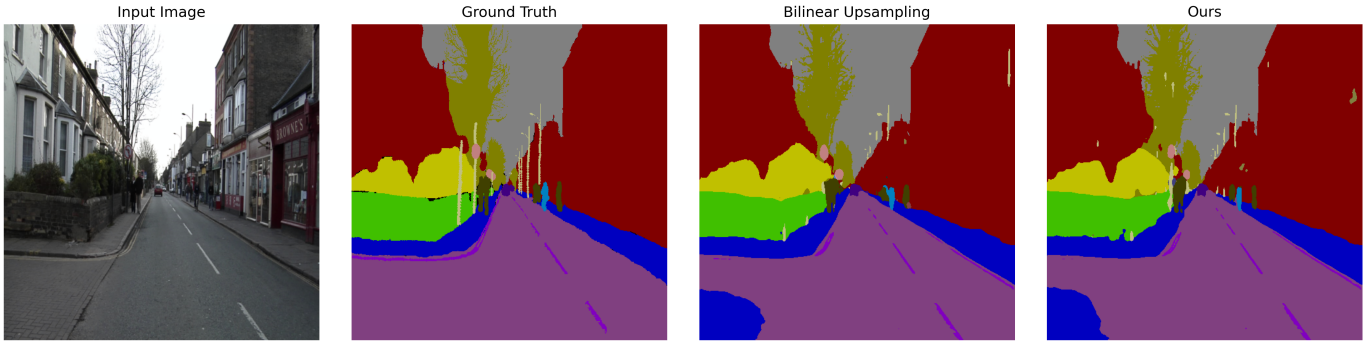

Fig. 8. Ours achieves better results on poles

every skip connection to pass through the guidance processing sub-network and the subsequent attention modules. This additional computation results in a reduction in inference speed (FPS) compared to simpler baselines. While this latency is acceptable for offline applications like medical image analysis, it poses challenges for deployment in ultra-low-latency environments, such as autonomous driving or real-time video processing, without further optimization.

Furthermore, the model exhibits a larger memory footprint during training. Because the guidance branch involves learnable convolutional layers, the network must store additional intermediate activation maps to compute gradients during backpropagation. This increases VRAM consumption, potentially constraining the maximum batch size on consumer-grade hardware. Researchers with limited computational resources might find it difficult to scale this architecture to very deep backbones (e.g., ResNet101 or larger) without employing techniques like gradient checkpointing or mixed-precision training.

Finally, the architecture's heavy reliance on the decoder for feature refinement creates an asymmetry that may not be ideal for mobile or edge deployment. Modern efficient networks often shift complexity to the encoder, keeping the decoder lightweight to save energy. Our approach does the opposite, placing significant computational weight on the upsampling path. Consequently, deploying this model on edge devices (such as mobile phones or embedded IoT systems) would likely require model compression techniques, such as pruning or the substitution of standard convolutions with depthwise-separable convolutions, to maintain acceptable performance.

5 Conclusion

This study set out to address a fundamental limitation in standard U-Net architectures: the ”semantic gap” that occurs when deep, abstract features are blindly merged with shallow, high-resolution details. Our solution was the introduction of the Feature-Guided Upsampling module, a novel decoder module that treats feature fusion as an active, intelligent process rather than a passive concatenation.

Our experiments lead to three distinct conclusions. First, by moving the attention mechanism to the post-fusion stage, we allowed the network to filter out the background noise that skip connections often introduce. This proved superior to the standard baseline in recovering fine geometric details, such as thin vertical poles, which typically vanish during the downsampling process.

Second, our model is robust and stable. A common fear in deep learning is that adding ”heavy” modules like attention requires delicate hyperparameter tuning. However, our results showed that the Feature-Guided Attention U-Net converged with the exact same stability as the simpler baseline. This suggests that our architecture actually simplifies the optimization task by mathematically aligning the encoder and decoder features before they meet, rather than forcing the network to learn difficult associations from scratch.

Finally, we demonstrated that decoder-side attention is sufficient. We achieved these gains using a standard, off-the-shelf ResNet34 backbone. This is a crucial finding for practical applications, as it proves that one does not need to retrain massive, custom encoders to achieve state-of-the-art segmentation results. Instead, focusing computational resources on the upsampling path – where the reconstruction actually happens – yields a more efficient and effective model.

Future work: Although our model improves upon baseline, further comparisons need to be made against other upsampling methods like transpose convolution or pixel shuffle. Hyperparameter optimization should also be performed to find out the maximum capabilities of the network. FGU's compatibility with other encoder backbones such as vision transformers can also be investigated.

References

[1] K. Muhammad, T. Hussain, H. Ullah, J. Del Ser, M. Rezaei, N. Kumar, M. Hijji, P. Bellavista, and V. H. C. De Albuquerque, “Vision-based semantic segmentation in scene understanding for autonomous driving: Recent achievements, challenges, and outlooks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 22 694–22 715, 2022.

[2] W. Kim and J. Seok, “Indoor semantic segmentation for robot navigating on mobile,” in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2018, pp. 22–25.

[3] S. Asgari Taghanaki, K. Abhishek, J. P. Cohen, J. Cohen-Adad, and G. Hamarneh, “Deep semantic segmentation of natural and medical images: a review,” *Artificial intelligence review*, vol. 54, no. 1, pp. 137–178, 2021.

[4] B. Neupane, T. Horanont, and J. Aryal, “Deep learning-based semantic segmentation of urban features in satellite images: A review and meta-analysis,” *Remote Sensing*, vol. 13, no. 4, p. 808, 2021.

[5] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[6] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[7] W. Xu, Y.-L. Fu, and D. Zhu, “Resnet and its application to medical image processing: Research progress and challenges,” *Computer Methods and Programs in Biomedicine*, vol. 240, p. 107660, 2023.

[8] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cbam: Convolutional block attention module,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[9] Y. Pang, Y. Li, J. Shen, and L. Shao, “Towards bridging semantic gap to improve semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4230–4239.

[10] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[11] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, “Attention u-net: Learning where to look for the pancreas,” *arXiv preprint arXiv:1804.03999*, 2018.

[12] A. Miglani, “Fga-net: Feature-gated attention for glioma brain tumor segmentation in volumetric mri images,” in *Artificial Intelligence and Knowledge Processing: 4th International Conference, AIKP 2024, Johannesburg Business School, Johannesburg, South Africa, August 22–24, 2024, Proceedings*, vol. 2228. Springer Nature, 2024, p. 66.

[13] G. Lin, A. Milan, C. Shen, and I. Reid, “Refinenet: Multi-path refinement networks for high-resolution semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[14] D. Mazzini, “Guided upsampling network for real-time semantic segmentation,” *arXiv preprint arXiv:1807.07466*, 2018.

[15] E. A. Brempong, S. Kornblith, T. Chen, N. Parmar, M. Minderer, and M. Norouzi, “Denoising pretraining for semantic segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4175–4186.

[16] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[17] S. Zagoruyko and N. Komodakis, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer,” *arXiv preprint arXiv:1612.03928*, 2016.

[18] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, “Segmentation and recognition using structure from motion point clouds,” in *European conference on computer vision*. Springer, 2008, pp. 44–57.